

# IMPLEMENTASI METODE SINGLE LINKAGE UNTUK MENENTUKAN KINERJA AGENT PADA CALL CENTRE BERBASIS ASTERISK FOR JAVA

Beni Ilham Priyambodo, Mike Yuliana, ST, MT, Nur Rosyid Mubtada'I, S. Kom

Jurusan Teknik Telekomunikasi - Politeknik Elektronika Negeri Surabaya

Institut Teknologi Sepuluh Nopember (ITS) Surabaya

Kampus PENS-ITS, Keputih, Sukolilo, Surabaya.

Telp : +62+031+5947280; Fax. +62+031+5946011

Email : [beni@student.eepis-its.edu](mailto:beni@student.eepis-its.edu), [mieke@eepis-its.edu](mailto:mieke@eepis-its.edu), [rosyid@eepis-its.edu](mailto:rosyid@eepis-its.edu)

---

## Abstrak

Keberadaan suatu *Call Centre* pada tiap-tiap provider telepon sangatlah diperlukan. Hal ini menyangkut kepuasan pelanggan terhadap produk suatu provider. *Call Centre* sendiri difungsikan agar pelanggan suatu provider telepon bisa berbicara langsung dengan *Agent (customer service)* dari operator telepon tersebut. Untuk itu diperlukan suatu kinerja yang baik dari seorang *Agent* pada *Call Centre*.

Pada tugas akhir ini, dibuat suatu sistem *Call Centre* menggunakan *Asterisk* berbasis pemrograman *Java*. Dimana digunakan suatu program *Java* untuk melakukan monitoring terhadap kinerja *Agent*. Dari hasil monitoring didapat lalu dilakukan pengelompokan menggunakan dua metode yaitu perhitungan manual dan metode *Single Linkage*. Pengelompokan manual dilakukan berdasar standarisasi yang telah ditetapkan, sedangkan pengelompokan menggunakan metode *Single Linkage* mengacu pada jarak terdekat antar parameter *Agent*.

Dari hasil pengujian didapatkan bahwa sistem *Call Centre* untuk menentukan kinerja *Agent* berbasis *Asterisk for Java* telah berjalan dengan baik. Waktu eksekusi paling lama yang dibutuhkan untuk menyimpan data monitoring yaitu parameter *Login Time* dengan rata-rata waktu 2.2 milisekon. Sedangkan untuk waktu eksekusi program clustering manual, semakin banyak jumlah *Agent* maka waktu eksekusi program semakin lama, dimana saat dilakukan pengujian terhadap 10 *Agent* dibutuhkan waktu 352.8 milisekon. Saat proses pengelompokan menggunakan metode manual dan metode *Single Linkage*, terdapat perbedaan dalam hal jumlah *Agent* dalam cluster, anggota pada masing-masing cluster, dan waktu eksekusi untuk melakukan cluster. Dalam hal waktu eksekusi, metode *Single Linkage* lebih unggul dengan waktu 184.7 ms dibanding metode manual 352.8 ms dalam mengelompokkan 10 *Agent*. Dengan mencari

nilai variance, didapat bahwa cluster yang terbentuk dari metode manual lebih ideal dibanding metode *Single Linkage*. Dimana nilai variance dari metode manual yaitu 1.4172645672185147 lebih kecil dibanding nilai variance metode *Single Linkage* yaitu 1.4568698006373213.

Kata kunci: *Call Centre*, *Java*, *Agent*, *Database*, monitoring, kinerja *Agent*, *Single Linkage*.

## 1. PENDAHULUAN

Pada era komunikasi yang begitu maju dan pesat, banyak orang yang menginginkan cara akses informasi yang cepat dan mudah. Salah satu upaya yang bisa dilakukan adalah dengan membuat *Call Centre*. Saat ini banyak perusahaan yang memiliki *Call Centre*, dimana *Call Centre* tersebut digunakan untuk beberapa hal yang antara lain meliputi pemberian informasi, serta penanganan kritik dan saran.

Tuntutan adanya *Call Centre* ini berimplikasi pada kebutuhan akan tenaga kerja operator/*Agent*, yang berguna untuk menangani panggilan yang masuk. Untuk menentukan kinerja dari operator *Call Centre* bisa dilakukan penilaian pada beberapa parameter yang didapat dari riset yang dilakukan oleh *Call Centre* Indosat, meliputi *N Inbound*, *Login Time*, *Handling Time*, dan *Time not Ready*. Parameter-parameter tersebut memiliki standarisasi tertentu yang nantinya digunakan untuk menentukan kinerja dari seorang *Agent*.

Pada tugas akhir ini, akan dibuat suatu sistem *Call Centre* menggunakan *Asterisk* berbasis pemrograman *Java*. Lalu di dalam sistem tersebut akan ditambahkan program untuk menentukan kinerja dari operator tersebut menggunakan metode *Single Linkage*. Sehingga kita bisa mengetahui kinerja dari *Agent* itu sendiri.

## 2. Dasar Teori

### 2.1 Call Centre

*Call Centre* merupakan suatu kantor informasi yang terpusat yang digunakan untuk tujuan menerima dan mengirimkan sejumlah besar permintaan melalui telepon. *Call Centre* dioperasikan oleh sebuah perusahaan sebagai pengadministrasi layanan yang mendukung produk incoming dan menyelidiki informasi tentang konsumen. Panggilan keluar *Call Centre* digunakan untuk telemarketing, *clientele*, dan debt collection pun dapat dilakukan. Sebagai tambahan untuk *Call Centre*, bahwa penanganan secara kolektif untuk surat, fax, dan email dalam sebuah lokasi lebih sering disebut dengan *contact center*.

### 2.2 Performansi Kinerja Agent pada Call Centre

Setiap Call Centre tentunya memiliki standarisasi sendiri untuk urusan penilaian kinerja Agent-nya. Standarisasi tersebut umumnya terdiri dari beberapa parameter penilaian diantaranya adalah :

- N Inbound

Jumlah panggilan yang diterima selama selang waktu tertentu.

$$N_{total} = N_{inbound} + N_{tolak} \dots\dots\dots (1)$$

Ket:

$N_{total}$  = jumlah seluruh panggilan yang masuk maupun ditolak

$N_{inbound}$  = jumlah panggilan yang diterima

$N_{tolak}$  = jumlah panggilan yang ditolak

- Login Time

Waktu yang dipergunakan oleh Agent untuk login. Cara penghitungannya yaitu:

$$Login = L_o - L_i - I_s \dots\dots\dots (2)$$

Ket:

$L_o$  = waktu logout

$L_i$  = waktu login

$I_s$  = lama waktu istirahat

- Handling Time

Waktu yang dibutuhkan oleh Agent untuk menerima panggilan. Jumlah rata-rata waktu yang dibutuhkan Agent untuk menerima panggilan masuk. Cara menghitungnya:

$$H_T = \sum h_t / N_{inbound} \dots\dots\dots (3)$$

Ket :

$H_T$  = handling time

$h_t$  = jumlah handling time masing-masing panggilan

$n_{inbound}$  = jumlah panggilan yang masuk

- Time not Ready

Waktu tiap-tiap Agent dalam keadaan tidak menerima panggilan (idle). Dihitung saat Agent selesai menerima panggilan hingga menerima panggilan kembali.

$$T_{idle} = T_{call} - T_{hangup} \dots\dots\dots (4)$$

Ket:

$T_{idle}$  = waktu idle

$T_{call}$  = waktu Agent menerima panggilan selanjutnya

$T_{hangup}$  = waktu Agent menutup panggilan sebelumnya

### 2.3 Asterisk Manager Interface (AMI)

*Asterisk Manager Interface* (AMI) adalah program *interface* yang handal. Program ini akan mengijinkan program lain untuk bisa mengontrol serta memonitor system *Asterisk*.

*Asterisk Manager Interface* menunggu koneksi dari port jaringan. Program dari *client* bisa terhubung dengan *Asterisk Manager Interface* pada port tersebut, melakukan autentikasi, dan mengirimkan perintah pada *Asterisk*. Lalu *Asterisk* akan merespon permintaan koneksi tersebut, serta meng-update program *client* dengan status dari sistem.

Untuk menggunakan fungsi dari *Manager*, dapat mendefinisikan *data account* pada file di */etc/Asterisk/Manager.conf*. User dapat terhubung dengan AMI pada port 5038 dari system dengan menggunakan fungsi telnet[1].

### 2.4 Asterisk

*Asterisk* merupakan salah satu *software* Server VoIP yang di distribusikan melalui GPL (*GNU General Public License*) yang artinya *Asterisk* adalah seperti *software opensource* lainnya yang bisa didownload secara grtis di internet. *Asterisk* serng disebut juga sebagai IPPBX, yaitu memiliki fungsi dan kemampuan layaknya PBX akan tetapi berbasis IP.

*Asterisk* merupakan implementasi dari telepon *Private Branch Exchange* (PBX) yang dibuat oleh Mark Spencer dari Digium.Inc pada tahun 1999. Seperti pada umumnya PBX, *asterisk* mengizinkan sejumlah telepon untuk membuat panggilan dan menghubungkan servis telepon lain termasuk *Public Switched Telephone Network* (PSTN).

### 2.5 Asterisk for Java

Paket *Asterisk for Java* terdiri dari satu set *class Java* yang memungkinkan untuk membuat aplikasi *Java* yang terintegrasi dengan *Asterisk PBX server*. *Asterisk for Java* mendukung masing – masing *interface* yang ditetapkan *Asterisk* untuk hal ini. Yaitu protokol FastAGI dan *Manager AGI*.

*Manager API* merupakan cara lain untuk berkomunikasi dengan server *Asterisk*. *Manager API* terdiri dari 3 konsep : *Action*, *Responses* dan *Events*. *Action* dapat dikirim ke *Asterisk* dan menginstruksikan untuk melakukan sesuatu.

Sambungan ke server *Asterisk* melalui *ManagerAPI* melalui TCP/IP pada port *default* 5038. Untuk mengaktifkan *ManagerAPI* pada

Asterisk, yaitu dengan mengganti *file* konfigurasi *Manager.conf* dan restart *Asterisk*. *Manager.conf* terkendala pada alamat IP yang diizinkan untuk terhubung, terdapat *username* serta *password* untuk otentifikasi[2].

## 2.6 Call Detail Records

*Call Detail Record* (CDR) pada teknologi voice over IP (VoIP) adalah suatu *file* informasi tentang penggunaan sistem seperti identitas pengguna, identitas tujuan pengguna, durasi panggilan, jumlah tagihan untuk tiap-tiap panggilan, total waktu penggunaan pada periode tagihan, total waktu koong pada periode tagihan, dan total pembayaran selama periode tagihan[3].

CDR berisi data-data tertentu yang berkaitan dengan panggilan-panggilan yang dilakukan. Data-data tersebut diantaranya[4].

- *Account code* : nomor *account* yang digunakan
- *src* : nomor identitas pemanggil
- *dst* : ekstensi tujuan, dll

## 2.7 Metode Single Linkage

*Single Linkage* merupakan salah satu contoh dari bentuk pengelompokan data (*clustering*). Metode pengelompokan ini menggunakan obyek yang paling dekat atau paling sama antar objek satu dengan yang lain untuk dikelompokkan.

Untuk menggunakan algoritma ini ada beberapa langkah yang harus dilakukan yaitu :

1. Mulai dengan *N cluster*, setiap *cluster* mengandung entiti tunggal dan sebuah matriks simetrik dari jarak (similarities) *D* = {dik} dengan tipe *NxN*.
2. Cari matriks jarak untuk pasangan *cluster* yang terdekat (paling mirip). Misalkan jarak antara *cluster U* dan *V* yang paling mirip adalah *duv*.
3. Gabungkan *cluster U* dan *V*. Label *cluster* yang baru dibentuk dengan (*UV*). Update entries pada matrik jarak dengan cara :
  - a. Hapus baris dan kolom yang bersesuaian dengan *cluster U* dan *V*
  - b. Tambahkan baris dan kolom yang memberikan jarak-jarak antara *cluster (UV)* dan *cluster-cluster* yang tersisa.
4. Ulangi langkah 2 dan 3 sebanyak (*N-1*) kali. (Semua objek akan berada dalam *cluster* tunggal setelah algoritma berakhir). Catat identitas dari *cluster* yang digabungkan dan tingkat-tingkat (jarak atau similaritas) di mana penggabungan terjadi[5].

Dalam pengerjaan menggunakan metode ini, ada beberapa perhitungan yang harus dilakukan diantaranya:

1. Mencari nilai rata-rata tiap parameter

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \dots \dots \dots (5)$$

Ket :

$\bar{x}$  = nilai rata-rata parameter

$x_i$  = nilai parameter

*n* = jumlah Agent

2. Mencari nilai standar deviasi tiap parameter

$$std(x) = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} \dots \dots \dots (6)$$

Ket :

std(x) = nilai standarisasi parameter

$x_i$  = nilai parameter

$\bar{x}$  = nilai rata-rata parameter

3. Mencari nilai standar untuk masing-masing Agent

$$z_i = \frac{x_i - \bar{x}}{std(x)} \dots \dots \dots (7)$$

Ket :

$z_i$  = nilai standar masing-masing Agent

$x_i$  = nilai parameter

$\bar{x}$  = nilai rata-rata parameter

std(x) = nilai standarisasi parameter

Setelah mencapai nilai normalisasi, maka dilakukan perhitungan menggunakan Euclidean Distance.

$$d_{rs} = (\sum_{i=1}^k (x_{ri} - x_{si})^2) \dots \dots \dots (8)$$

Ket :

*r* = Agent ke 1-9, dimana *s* = Agent ke *r*+1

*k* = jumlah Agent

$d_{rs}$  = jarak antara Agent *r* dengan *s*

$x_{ri}$  = nilai parameter Agent *r* ke-1 s/d 9

$x_{si}$  = nilai parameter Agent *r*+1

Untuk memberikan label pada cluster yang ditentukan sesuai dengan standarisasi yang ada, maka digunakan rumus berikut.

$$total = \sqrt{x_a^2 + x_b^2} \dots \dots \dots (9)$$

Ket :

total = nilai akhir

$x_a$  = nilai parameter pertama

$x_b$  = nilai parameter kedua

## 2.8 Cluster Analysis (Variance)

Digunakan untuk mengukur nilai hasil penyebaran data-data hasil clustering. Variance pada clustering ada dua macam yaitu:

- a. Variance within cluster : Tipe varian ini mengacu pada jarak antar anggota pada cluster yang sama
- b. Variance between cluster : Tipe varian ini mengacu pada jarak antar cluster

Ada dua ketentuan apabila menentukan cluster ideal menggunakan cara perbandingan Variance within Cluster ( $V_w$ ) dan Variance between Cluster ( $V_b$ ) yaitu sebagai berikut.

- a. Berdasar nilai minimum

$$V = \frac{V_w}{V_b} \dots \dots \dots (10)$$

Ket :

*V* = nilai variance

$V_w$  = nilai variance within cluster

$V_b$  = nilai variance between cluster

Cluster yang disebut ideal adalah cluster yang memiliki nilai variance yang paling kecil.

b. Berdasar nilai maksimum

$$V = \frac{V_B}{V_W} \dots\dots\dots (11)$$

Ket :

V = nilai variance

$V_W$  = nilai variance within cluster

$V_B$  = nilai variance between cluster

Cluster yang disebut ideal adalah cluster yang memiliki nilai variance yang paling besar.

Sebelum mencari nilai variance (V), perlu dicari nilai variance within cluster ( $V_W$ ) dan nilai variance between cluster ( $V_B$ )[6].

a. Variance within Cluster ( $V_W$ )

$$V_W = \frac{1}{N-k} \sum_{i=1}^k (n_i - 1) \cdot V_i^2 \dots\dots\dots (12)$$

Ket :

N = jumlah semua data

k = jumlah cluster

$n_i$  = jumlah data pada cluster ke-i

$V_i^2$  = variance pada cluster ke-i

Sebelum menghitung variance within perlu menghitung nilai  $V_i^2$ .

$$V_c^2 = \frac{1}{n_c - 1} \sum_{i=1}^{n_c} (d_i - \bar{d}_i)^2 \dots\dots\dots (13)$$

Ket :

$V_c^2$  = variance pada cluster c

c = 1...k, dimana k = jumlah cluster

$n_c$  = jumlah data pada cluster c

$d_i$  = data ke-i pada suatu cluster

$\bar{d}_i$  = rata-rata dari data pada suatu cluster

b. Variance between Cluster ( $V_B$ )

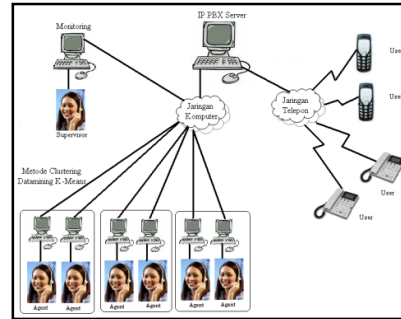
$$V_B = \frac{1}{k-1} \sum_{i=1}^k n_i (\bar{d}_i - \bar{d})^2 \dots\dots\dots (14)$$

Ket :

$\bar{d}$  = rata-rata dari  $\bar{d}_i$

### 3. Metodologi

“Implementasi Metode *Single Linkage* untuk Menentukan Kinerja Agent pada *Call Centre* Berbasis *Asterisk for Java*” akan memberikan kemudahan bagi suatu *Call Centre* untuk dapat menentukan kinerja dari *Agent-Agent* mereka. Karena disini sistem telah terintegrasi dengan aplikasi untuk menghitung performansi dari kinerja *Agent-Agent* *Call Centre*. Ilustrasi dari sistem Implementasi Metode *Single Linkage* untuk Menentukan Kinerja Agent pada *Call Centre* Berbasis *Asterisk for Java* dapat dilihat pada gambar di bawah.



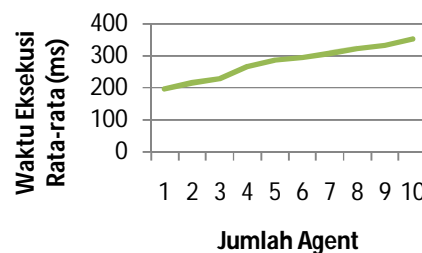
**Gambar 1.**Ilustrasi sistem Implementasi Metode *Single Linkage* untuk Menentukan Kinerja Agent pada *Call Centre* Berbasis *Asterisk for Java*

Cara kerja dari sistem ini adalah saat ada panggilan masuk dan diterima oleh *Agent*, maka panggilan tersebut akan dicatat dan disimpan pada *Database*. Data yang disimpan berupa jumlah panggilan yang dihitung tiap panggilan masuk yang diterima *Agent*, rata-rata lama panggilan yang digunakan oleh *Agent*, lama waktu *idle Agent*, dan lama waktu *Login* dari *Agent* terhadap *server*. Data-data tersebut disimpan dalam tabel yang nanti bisa dilihat setiap saat. Untuk melakukan penilaian performansi dari kinerja *Agent*, digunakan sebuah metode *clustering* yaitu *Single Linkage* yang dirancang menggunakan pemrograman *Java* dengan cara mengelompokkan *Agent-Agent* tersebut dalam beberapa kategori penilaian.

### 4. Pembahasan dan Analisa

#### 4.1.Waktu Eksekusi Clustering Manual

Perhitungan kriteria Agent dilakukan terhadap 10 Agent. Untuk melakukan perhitungan dibutuhkan program tambahan dimana untuk menghitung waktu eksekusi untuk perhitungan jumlah Agent bermacam-macam.



**Gambar 2.** Grafik Waktu Eksekusi Clustering Manual

Dari gambar 3 didapat bahwa semakin banyak jumlah Agent yang dihitung, maka semakin lama waktu eksekusi yang dibutuhkan untuk menentukan kinerja Agent. Hal ini disebabkan banyaknya program yang harus dikerjakan untuk menghitung nilai tiap parameternya.

#### 4.2. Clustering Manual

Setelah parameter-parameter didapat, dilakukan proses pengelompokan secara manual (tidak menggunakan metode) dengan menggunakan standart penilaian yang ditetapkan oleh Call Centre pada umumnya. Ada 3 Kriteria penilaian Agent pada proyek akhir ini berdasarkan produktifitas Agent. Penghitungan produktifitas Agent berdasar jumlah panggilan dan waktu login yang dihitung selama 1 bulan.

Pengelompokan Agent tersebut dilakukan dengan mengacu pada standarisasi yang ditetapkan untuk kedua parameter produktifitas. Untuk parameter jumlah panggilan per bulan digunakan standart penilaian sebagai berikut.

$$\begin{aligned} n > 340 &= 100 \\ 280 < n \leq 340 &= 70 \\ n \leq 280 &= 50 \end{aligned}$$

dimana :  
n = jumlah panggilan

Sedangkan untuk standarisasi parameter Login Time berdasar standart penilaian dibawah dimana waktu telah dikonversi menjadi detik selama 1 bulan.

$$\begin{aligned} t > 54000 &= 100 \\ 396000 < t \leq 54000 &= 70 \\ t \leq 396000 &= 50 \end{aligned}$$

dimana :  
t = jumlah waktu (detik)

Setelah dilakukan penilaian sesuai dengan standarisasi sebelumnya, dicari nilai total dari kedua parameter dengan rumus sebagai berikut.

$$x = (n + t) \div 2 \dots\dots\dots (15)$$

dimana x adalah nilai total.

Dari nilai total yang didapat, lalu dilakukan pengelompokan Agent dengan standarisasi akhir sebagai berikut.

$$\begin{aligned} x > 75.5 &= \text{baik} \\ 50 < x \leq 75.5 &= \text{sedang} \\ x \leq 50 &= \text{buruk} \end{aligned}$$

Dengan menggunakan standarisasi yang sudah ada, dengan data masing-masing Agent yang telah ada, dilakukan pengelompokan kinerja berdasarkan produktifitas[7].

**Tabel 1.** Penilaian Kinerja Agent

Agent	n	t	x	Nilai
Hasbi	2230	492031	70	Baik
Dipta	2274	513678	80	Baik sekali
Beni	1842	498105	60	Kurang
Tasya	2650	527757	90	Baik sekali
Daus	2217	490861	80	Baik sekali
Ubai	2430	501095	80	Baik sekali
Nanda	1745	437976	50	Buruk
Khusnul	1732	445545	50	Buruk
Dani	1750	437828	70	Baik
Jai	2000	470000	70	Baik

#### 4.3. Perbandingan Hasil Clustering Metode Manual dan Metode Single Linkage

Perbandingan hasil cluster antara metode manual dan Single Linkage dilakukan dengan 3 cara yaitu:

##### 1. Waktu eksekusi

Untuk membagi Agent ke 3 cluster yang telah ditentukan menggunakan program yang ada, dibutuhkan waktu. Berikut adalah pengukuran waktu eksekusi yang dilakukan sebanyak 10 kali saat melakukan proses cluster menggunakan metode Single Linkage.

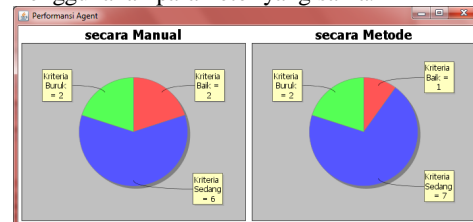
**Tabel 2.** Waktu Eksekusi Program Single Linkage

Urutan Eksekusi	Waktu (ms)
1	176
2	179
3	173
4	219
5	201
6	168
7	162
8	179
9	194
10	196
Rata-rata	184.7

Dari hasil pengukuran waktu eksekusi sebanyak 10 kali, didapat hasil rata-rata waktu eksekusi sebesar 184.7 ms. Jika dibandingkan dengan waktu eksekusi metode manual saat kondisi Agent sebanyak 10 pada gambar 4.8, waktu eksekusi dengan menggunakan metode Single Linkage relatif lebih cepat. Hal ini dikarenakan saat menggunakan metode manual untuk proses cluster dilakukan pengecekan parameter terhadap standarisasi untuk tiap-tiap Agent, sehingga otomatis membutuhkan waktu lebih karena melakukan proses yang sama berulang-ulang. Sedangkan saat menggunakan Single Linkage tidak terjadi pengecekan parameter terhadap standarisasi untuk tiap Agentnya.

##### 2. Jumlah anggota masing-masing cluster

Dari hasil pengelompokan menggunakan metode Single Linkage, telah didapatkan 3 cluster sesuai dengan kriteria Agent yang telah ditetapkan. Dengan melihat hasil antara clustering manual dan clustering Single Linkage, disimpulkan terjadi perbedaan jumlah anggota cluster meskipun diproses menggunakan parameter yang sama.



**Gambar 3.** Perbandingan Jumlah Agent dalam Cluster secara Manual dan Metode Single Linkage

### 3. Anggota dari masing-masing cluster

Selain terdapat perbedaan jumlah anggota cluster, juga didapat bahwa terjadi perbedaan pada anggota di masing-masing cluster.



No	Nama Agent	N inbound	Login Time	Cluster Manual	Single Linkage
1	Hasbi	335	140:15:21	sedang	sedang
2	ais	326	142:41:18	sedang	sedang
3	Benti	348	139:47:26	baik	sedang
4	Tasya	360	146:35:57	baik	sedang
5	Daus	269	147:27:41	sedang	baik
6	asri	303	139:11:35	sedang	sedang
7	ayu	260	90:49:36	buruk	buruk
8	Khurnul	290	123:45:45	sedang	sedang
9	Dani	328	149:23:48	sedang	sedang
10	Jai	276	100:0:0	buruk	buruk

**Gambar 4.** Hasil Eksekusi Performansi Agent

Perbedaan tersebut diakibatkan saat menggunakan metode manual, maka anggota cluster ditentukan menggunakan standarisasi yang telah ditetapkan untuk tiap parameter. Perhitungan yang dilakukan akan lebih akurat karena menyesuaikan standarisasi. Namun saat menggunakan metode Single Linkage, maka standarisasi tiap parameter tidak akan dipakai untuk menentukan proses cluster. Akan tetapi menggunakan jarak terdekat antar Agent yang dicari menggunakan rumus Euclidean Distance.

#### 4.4. Menentukan Cluster yang Ideal Menggunakan Variance

Dari hasil eksekusi program clustering menggunakan metode manual dan Single Linkage, didapat beberapa perbedaan diantaranya dari segi waktu eksekusi, jumlah anggota dalam cluster, dan anggota dalam cluster tersebut. Untuk menentukan cluster yang ideal antara cluster dari metode manual dan metode Single Linkage, digunakan nilai variance.

Untuk mendapat nilai variance dari suatu cluster, yang harus dihitung yaitu nilai variance between cluster dan variance within cluster. Dengan menggunakan rumus 12 dan rumus 14 bisa didapat nilai tersebut. Masing-masing hasil cluster dari kedua metode dihitung untuk mendapatkan nilai  $V_B$  dan  $V_W$ . Setelah didapat kedua nilai tersebut maka untuk mendapat hasil akhir berupa nilai variance digunakan rumus 10. Karena menggunakan nilai variance minimum, maka metode yang memiliki nilai variance paling kecil ditentukan sebagai metode yang memiliki cluster yang ideal. Dalam hal ini setelah dihitung, nilai variance dari metode manual yaitu 1.4172645672185147 dan nilai variance dari metode Single Linkage adalah 1.4568698006373213. maka ditetapkan cluster yang terbentuk dari penggunaan metode manual lebih ideal dibandingkan cluster yang dibentuk menggunakan metode Single Linkage.

### 5. Kesimpulan

Berdasarkan hasil analisa dan pembahasan, diperoleh kesimpulan sebagai berikut :

1. Untuk login, seorang manager harus menggunakan username dan password yang sesuai dengan yang tercatat pada file manager.conf, dimana penggunaan username tidak terlalu memperhatikan penggunaan huruf besar dan kecil, sedangkan penggunaan password harus sesuai baik kata maupun penggunaan huruf besar dan kecil, karena untuk password berlaku case sensitif.
2. Waktu eksekusi paling lama untuk menyimpan data monitoring berasal dari parameter Login Time dengan lama waktu 2.2 milisekon. Penyebab waktu eksekusi penyimpanan Login Time paling lama adalah karena kondisi sistem yang baru berjalan dan butuh waktu untuk mengecek username serta password yang dimasukkan Agent.
3. Kelebihan metode manual yaitu keakuratan pembentukan cluster dibanding dengan metode Single Linkage dilihat dari perhitungan nilai variance. Dimana nilai variance yang didapat yaitu 1.4172645672185147 lebih kecil dibandingkan nilai variance metode Single Linkage 1.4568698006373213.
4. Kekurangan dari metode manual adalah waktu eksekusi yang cenderung lama karena saat melakukan cluster akan terus melakukan pengecekan parameter Agent dengan standarisasi. Untuk melakukan cluster terhadap 10 Agent dibutuhkan waktu 352.8 ms. Sedangkan metode Single Linkage membutuhkan waktu eksekusi yang relatif cepat yaitu sebesar 184.7 ms, karena dalam proses pembentukan cluster tidak perlu melakukan pengecekan berulang-ulang terhadap standarisasi parameter.

### DAFTAR PUSTAKA

- [1] Fitriyadi. *Implementasi Screen Pop pada Layanan informasi Mahasiswa Berbasis CTI*. Surabaya. (2010)
- [2] <http://asterisk-java.org/tutorial>
- [3] Goncalves, Flavio E. *Asterisk PBX 3<sup>rd</sup> Generation*. (2007)
- [4] Goncalves, Flavio E. *Configuration Guide for Asterisk<sup>TM</sup> PBX*. (2007)
- [5] Hartini, Entin. *Modul Ajar Metode Clustering Hirarki*. Surabaya. PENS-ITS.
- [6] Ridho Barakbah, Ali. *Modul ajar Cluster Analysis*. Surabaya. PENS-ITS.
- [7] Malcolm Carlaw, Peggy Carlaw, Vasudha K. Deming, Kurt Friedmann, *Managing and Motivating Contact Center Employees*. (2000)